

This listing of claims will replace all prior versions, and listings, of claims in the application:

**LISTING OF THE CLAIMS:**

1. (Original) A method for routing a datagram that has been fragmented into a plurality of fragments utilizing content-based routing information included in one or more fragments of the plurality of fragments, the method comprising:

(a) generating a context for the datagram associated with routing the plurality of fragments of the datagram and setting the context for the datagram to passive until content-based routing information included in the one or more fragments is received;

(b) caching received fragments while the context is set passive;

(c) determining a destination for routing the plurality of fragments when content-based routing information included in the one or more fragments is received and setting the context for the datagram to active; and

(d) routing any cached fragments and subsequently received fragments of the datagram to the determined destination while the context is active without reassembling the plurality of fragments into the datagram.

2. (Original) The method according to Claim 1, wherein the routing step routes the subsequently received fragments without caching the subsequently received fragments.

3. (Original) The method according to Claim 1, wherein the content-based routing information is included in a sequentially first fragment of the datagram.

4. (Original) The method according to Claim 3, wherein the routing step further comprises routing the sequentially first fragment to the determined destination before the any cached fragments.

5. (Original) The method according to Claim 1, wherein the routing step further comprises routing a sequentially last fragment of the plurality of fragments of the datagram to the

determined destination after all other of the plurality of fragments have been routed to the determined destination.

6. (Original) The method according to Claim 5, the method further comprising a step of tracking an aggregate length of the plurality of fragments by incrementing a fragment byte counter by a data length of each fragment of the datagram.

7. (Original) The method according to Claim 6, the method further comprising a step of identifying whether the plurality of fragments of the datagram have been processed, the identifying step comprising:

calculating a total length of the sequentially last fragment by adding its fragment offset and its data length; and

comparing the aggregate length against the total length of the sequentially last fragment.

8. (Original) The method according to Claim 1, the method further comprising a step of starting a timer upon generating the context for identifying a time period that the datagram is allowed to be on a network.

9. (Original) The method according to Claim 8, the method further comprising a step of discarding any cached fragments upon expiration of the timer.

10. (Original) The method according to Claim 1, the method further comprising a step of discarding any cached fragments if the content-based routing information is not received.

11. (Original) The method according to Claim 1, the method further comprising a step of providing a fragment queue associated with the context for the datagram for caching the fragments.

12. (Original) The method according to Claim 1, wherein the content-based routing information consists of one or more of the following:

one or more fields of an Internet Protocol (IP) header of the datagram;

one or more fields of the Transmission Control Protocol (TCP) header of the datagram;

one or more Universal Resource Locators (URLs) of the datagram; and

one or more cookies of the datagram.

13. (Original) A system for routing a datagram that has been fragmented into a plurality of fragments utilizing content-based routing information included in one or more fragments of the plurality of fragments, the system comprising:

(a) a receiving mechanism for receiving the plurality of fragments of the datagram;

(b) a control mechanism for generating a context for the datagram associated with routing the plurality of fragments of the datagram and setting the context for the datagram to passive until content-based routing information included in the one or more fragments is received;

(c) a cache for caching received fragments while the context is set to passive;

(d) a routing mechanism for determining a destination for routing the plurality of fragments when content-based routing information included in the one or more fragments is received and setting the context for the datagram to active; and

(e) a forwarding mechanism for transmitting any cached fragments and subsequently received fragments of the datagram to the determined destination while the context is active without reassembly of the plurality of fragments into the datagram.

14. (Original) The system according to Claim 13, wherein the subsequently received fragments are transmitted to the determined destination without being cached.

15. (Original) The system according to Claim 13, wherein the content-based routing information is included in a sequentially first fragment of the datagram.

16. (Original) The system according to Claim 15, wherein the sequentially first fragment is transmitted to the determined destination before the any cached fragments.

17. (Original) The system according to Claim 13, wherein a sequentially last fragment of the plurality of fragments of the datagram is transmitted to the determined destination after all other of the plurality of fragments have been transmitted to the determined destination.

18. (Original) The system according to Claim 17, an aggregate length of the plurality of fragments is tracked by incrementing a fragment byte counter by a data length of each fragment of the datagram.

19. (Original) The system according to Claim 18, wherein the system further comprises an a mechanism for identifying whether the plurality of fragments of the datagram have been processed, the identifying mechanism comprising:

- calculating a total length of the sequentially last fragment by adding its fragment offset and its data length; and

- comparing the aggregate length against the total length of the sequentially last fragment.

20. (Original) The system according to Claim 13, wherein a timer is started upon generating the context for identifying a time period that the datagram is allowed to be on a network.1.

21. (Original) The system according to Claim 20, wherein any cached fragments are discarded upon expiration of the timer.

22. (Original) The system according to Claim 13, wherein any cached fragments are discarded if the content-based routing information is not received.

23. (Original) The system according to Claim 13, wherein a fragment queue associated with the context for the datagram is provided for caching the fragments.

24. (Original) The system according to Claim 1, wherein the content-based routing information consists of one or more of the following:

- one or more fields of an Internet Protocol (IP) header of the datagram;

one or more fields of the Transmission Control Protocol (TCP) header of the datagram;

one or more Universal Resource Locators (URLs) of the datagram; and

one or more cookies of the datagram.

25. (Original) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps for routing a datagram that has been fragmented into a plurality of fragments utilizing content-based routing information included in one or more fragments of the plurality of fragments, the method comprising:

(a) generating a context for the datagram associated with routing the plurality of fragments of the datagram and setting the context for the datagram to passive until content-based routing information included in the one or more fragments is received;

(b) caching received fragments while the context is set to passive;

(c) determining a destination for routing the plurality of fragments when content-based routing information included in the one or more fragments is received and setting the context for the datagram to active; and

(d) routing any cached fragments and subsequently received fragments of the datagram to the determined destination while the context is active without reassembling the plurality of fragments into the datagram.

26. (Original) The program storage device according to Claim 25, wherein the routing step routes the subsequently received fragments without caching the subsequently received fragments.

27. (Original) The program storage device according to Claim 25, wherein the content-based routing information is included in a sequentially first fragment of the datagram.

28. (Original) The program storage device according to Claim 27, wherein the routing step further comprises routing the sequentially first fragment to the determined destination before the any cached fragments.

29. (Original) The program storage device according to Claim 25, wherein the routing step further comprises routing a sequentially last fragment of the plurality of fragments of the

datagram to the determined destination after all other of the plurality of fragments have been routed to the determined destination.

30. (Original) The program storage device according to Claim 29, the method further comprising a step of tracking an aggregate length of the plurality of fragments by incrementing a fragment byte counter by a data length of each fragment of the datagram.

31. (Original) The program storage device according to Claim 30, the method further comprising a step of identifying whether the plurality of fragments of the datagram have been processed, the identifying step comprising:

calculating a total length of the sequentially last fragment by adding its fragment offset and its data length; and

comparing the aggregate length against the total length of the sequentially last fragment.

32. (Original) The program storage device according to Claim 25, the method further comprising a step of starting a timer upon generating the context for identifying a time period that the datagram is allowed to be on a network.

33. (Original) The program storage device according to Claim 32, the method further comprising a step of discarding any cached fragments upon expiration of the timer.

34. (Original) The program storage device according to Claim 25, the method further comprising a step of discarding any cached fragments if the content-based routing information is not received.

35. (Original) The program storage device according to Claim 25, the method further comprising a step of providing a fragment queue associated with the context for the datagram for caching the fragments.

36. (Original) The program storage device according to Claim 25, wherein the content-based routing information consists of one or more of the following:

one or more fields of an Internet Protocol (IP) header of the datagram;  
one or more fields of the Transmission Control Protocol (TCP) header of the datagram;  
one or more Universal Resource Locators (URLs) of the datagram; and  
one or more cookies of the datagram.

37. (Original) A router for routing a datagram that has been fragmented into a plurality of fragments utilizing content-based routing information included in one or more fragments of the plurality of fragments, the router comprising:

- (a) a receiving mechanism for receiving the plurality of fragments of the datagram;
- (b) a control mechanism for generating a context for the datagram associated with routing the plurality of fragments of the datagram and setting the context for the datagram to passive until content-based routing information included in the one or more fragments is received;
- (c) a cache for caching received fragments while the context is set to passive;
- (d) a routing mechanism for determining a destination port for routing the plurality of fragments when content-based routing information included in the one or more fragments is received and setting the context for the datagram to active; and
- (e) a forwarding mechanism for transmitting any cached fragments and subsequently received fragments of the datagram to the determined destination port while the context is active without reassembly of the plurality of fragments into the datagram.

38. (Original) A server load balancer for routing a datagram that has been fragmented into a plurality of fragments utilizing content-based routing information included in one or more fragments of the plurality of fragments, the router comprising:

- (a) a receiving mechanism for receiving the plurality of fragments of the datagram;
- (b) a control mechanism for generating a context for the datagram associated with routing the plurality of fragments of the datagram and setting the context for the datagram to passive until content-based routing information included in the one or more fragments is received;

- (c) a cache for caching received fragments while the context is set to passive;
- (d) a routing mechanism for determining a destination device driver for routing the plurality of fragments when content-based routing information included in the one or more fragments is received and setting the context for the datagram to active; and
- (e) a forwarding mechanism for transmitting any cached fragments and subsequently received fragments of the datagram to the determined destination device driver while the context is active without reassembly of the plurality of fragments into the datagram.

39. (Original) A method for routing a datagram that has been fragmented into a plurality of fragments utilizing content-based routing information included in one or more fragments of the plurality of fragments, the method comprising:

- (a) generating a context for the datagram associated with routing the plurality of fragments of the datagram;
- (b) determining a destination for routing the plurality of fragments when content-based routing information included in the one or more fragments is received and setting the context for the datagram to active; and
- (d) routing subsequently received fragments of the datagram to the determined destination while the context is active without reassembling the plurality of fragments into the datagram.